

### **Remarks**

This Request for Continued Examination and Reply are in response to the Office Action mailed September 26, 2008, and to the Advisory Action mailed November 21, 2008.

#### **I. Summary of Examiner's Rejections**

In the Office Action mailed September 26, 2008, the Examiner rejected Claims 1, 3, 5-10, 12, 14-20, 22, 24, 26-31 and 33 under 35 U.S.C. 103(a) as being unpatentable over Kim et al. (U.S. Patent Publication No. 2003/0120686 A1, hereafter Kim), in view of Emmanuel Tanyi ("Easy XML," 2000, hereafter Tanyi), and further in view of Park et al. (U.S. Patent Publication No. 2004/0024812 A1, hereafter Park). Claims 11, 21 and 32 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kim, in view of Tanyi, further in view of Park, and further in view of IBM TDB ("Method and System for Visually Constructing Document Type Definitions and Related Artifacts Using a Reusable Object Model," 2001, hereafter IBM TDB).

#### **II. Summary of Applicant's Amendment**

The present Reply amends Claims 1, 5-11, 22, and 26-33; adds Claim 45; and cancels Claims 12 and 14-21, leaving for the Examiner's present consideration Claims 1, 3, 5-11, 22, 24, 26-32, and 45. Reconsideration of the Application, as amended, is respectfully requested.

#### **III. Claim Rejections under 35 U.S.C. §103(a)**

In the Office Action mailed September 26, 2008, the Examiner rejected Claims 1, 3, 5-10, 12, 14-20, 22, 24, 26-31 and 33 under 35 U.S.C. 103(a) as being unpatentable over Kim (U.S. Patent Publication No. 2003/0120686), in view of Tanyi ("Easy XML," 2000), and further in view of Park (U.S. Patent Publication No. 2004/0024812).

#### **Claim 1**

Claim 1 has been amended to more clearly define the embodiment therein. As amended, Claim 1 defines:

- 1. (Currently Amended) An interactive tool for viewing and manipulating a virtual content repository (VCR) having an application program interface (API), comprising:  
providing a first graphical user interface (GUI), having a navigation pane and an editor window, configured to present a hierarchical namespace, shown in the navigation pane, that spans information in a plurality of content repositories represented by the*

*virtual content repository, wherein the namespace includes at least one element, and wherein one of the at least one element can be selected;*

*displaying a content editor in the editor window when a content mode is selected in the navigation pane wherein the content editor is configured to present and to enable editing of content associated with the selected element in the navigation pane;*

*displaying a schema editor in the editor window when a type mode is selected in the navigation pane wherein the schema editor is configured to present and to enable editing of schema associated with the selected element in the navigation pane;*

*wherein the VCR logically represents the plurality of content repositories as a single content repository;*

*wherein each one of the plurality of content repositories represented by the VCR implements a different service provider interface (SPI) compatible with the API, wherein each SPI and the API share a content model, and wherein each SPI maps operations on the content model to operations on the content repository implementing that SPI; and*

*wherein the SPI enables each one of the plurality of content repositories to plug into the VCR.*

Claim 1, as amended, defines an interactive tool for viewing and manipulating a virtual content repository (VCR) having an application program interface (API). The interactive tool comprises providing a first graphical user interface (GUI), having a navigation pane and an editor window, configured to present a hierarchical namespace, shown in the navigation pane, that spans information in a plurality of content repositories represented by the virtual content repository, wherein the namespace includes at least one element, and wherein one of the at least one element can be selected; displaying a content editor in the editor window when a content mode is selected in the navigation pane wherein the content editor is configured to present and to enable editing of content associated with the selected element in the navigation pane; displaying a schema editor in the editor window when a type mode is selected in the navigation pane wherein the schema editor is configured to present and to enable editing of schema associated with the selected element in the navigation pane; wherein the VCR logically represents the plurality of content repositories as a single content repository; wherein each one of the plurality of content repositories represented by the VCR implements a different service provider interface (SPI) compatible with the API, wherein each SPI and the API share a content model, and wherein each SPI maps operations on the content model to operations on the content repository implementing that SPI; and wherein the SPI enables each one of the plurality of content repositories to plug into the VCR.

Kim discloses techniques for extensible stylesheet designs using meta-tag and/or

associated meta-tag information. (Abstract). Kim further discloses that upon receiving an XML file as a source document, the file is loaded into an XML parser such as Xerces-C++ from www.apache.org. Within the Xerces-C++, the source document is parsed as elements, attributes, and data in a tree-like structure that can be individually accessible via Xerces-C++ API. Within the XML parser, a user can select text data and attributes of the source XML document as associated meta-tag information and create meta-tag information for each associated meta-tag information with DOPI. (Paragraph [0081]).

Tanyi discloses a tool to rapidly create, validate, manipulate, and transform XML structures from ODBC datasources (e.g. Microsoft Access, Oracle, and MS SQL Server) or from URLs and local files. (Page 1). Easy XML offers a simple user interface that represents two views of your XML data. By providing a text view and a tree view, Easy XML gives you the possibility of doing minor changes like adding DTD/Schema, URLs, and processing instruction with the text view, and using the tree view for XML document structure manipulation and transformation. (Page 1).

Park discloses a content publication system supporting real-time integration and processing of multimedia content including dynamic data. (Abstract). The service publication server 4 basically includes both a service operating function and a service configuring function. (Paragraph [0029]). The service configuring function includes configuring various content to be included in a service in internal expression, i.e., XML, storing them in repository 8, and publishing the content stored in the repository 8 as a program. (Paragraph [0030]). The operating function includes publishing content stored in the repository 8 in real time at a users request. Basically, the repository 8 may store data produced by a service producer in advance (hereinafter referred to as static data) and data brought in from various sources in real time (hereinafter referred to as dynamic data. (Paragraph [0031]).

As described above Kim and Tanyi appear to disclose XML editors. Kim also appears to disclose that the XML files can be linked to HTML files. Park appears to disclose a system that enables static and dynamic content to be produced in response to users' requests.

Claim 1, as amended, defines a first graphical user interface (GUI), having a navigation pane and an editor window. Claim 1, as amended, further defines displaying a content editor in the editor window when a content mode is selected in the navigation pane; and displaying a schema editor in the editor window when a type mode is selected in the navigation pane.

Applicant respectfully submits that Kim, in view of Tanyi and Park, does not appear to disclose these features. Instead, Kim, in view of Tanyi and Park, appears to disclose separate and unrelated GUIs. For example, Figure 7C of Kim was cited as disclosing a second GUI,

however, Figure 7C shows two separate applications, each having their own GUIs: Microsoft's FrontPage and XMLCities' WebGenieXML. Further, Tanyi appears to disclose another application having its own GUI, separate and unrelated from the other cited GUIs. Applicant respectfully submits that Kim in view of Tanyi and Park does not disclose or render obvious the embodiment of Claim 1.

Furthermore, Claim 1 has been amended to more clearly define that each one of the plurality of content repositories represented by the VCR implements a service provider interface (SPI) compatible with the API, wherein each SPI and the API share a content model. Applicant respectfully submits that none of the cited references disclose that each SPI and the API share a content model.

Claim 1 has also been amended to more clearly define that each SPI maps operations on the content model to operations on the content repository implementing that SPI. Since Park only appears to disclose a single content repository, operations on the content repository can be tailored for that repository. However, the embodiment of Claim 1 defines a plurality of content repositories. This feature accounts for differences between the implementations of the plurality of content repositories.

It was asserted in the Office Action that Park discloses "a content repository, which includes a SPI compatible with an API." Claim 1, however, defines a *plurality* of content repositories represented by the VCR, and that *each one* of the plurality of content repositories implements a different service provider interface (SPI) compatible with the API. Applicant respectfully submits that Kim, in view of Tanyi and Park, does not disclose or render obvious these features.

Additionally, it was asserted in the Advisory Action that the service publication server, disclosed in Park, "provides for the same functionality of an SPI." Applicant respectfully disagrees. Park discloses that "the dynamic search/comparison service 1b provides an integrate search service for integrating data from various data sources and allowing for search based on search conditions." Thus, Park appears to disclose searching for data from a variety of data sources. Claim 1, however, clearly defines that the SPI enables each of the plurality of content repositories to plug into the VCR. Thus, in Claim 1, the data stored in each content repository is not merely searched, but instead each content repository is plugged into the VCR. Since Park searches only data and does not appear to integrate data sources, there appears to be no need for those data sources to implement an SPI as defined in Claim 1. Applicant respectfully submits that Kim, in view of Tanyi and Park, does not disclose an SPI, as defined in Claim 1.

In view of the above comments, Applicant respectfully submits that Claim 1, as currently amended, is neither anticipated by nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 22 and 33**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Claims 22 and 33 have been similarly amended to more clearly define the embodiments therein. For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claims 22 and 33, as amended, are likewise neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 3, 5-10, 24, and 26-31**

Claims 3, 5-10, 24, and 26-31 depend from and include all of the features of Claims 1 or 22. Claims 3, 5-10, 24, and 26-31 have not been addressed separately but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Reconsideration thereof is respectfully requested.

#### **Claims 11, 21, and 32**

In the Office Action mailed September 26, 2008, Claims 11, 21 and 32 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kim, in view of Tanyi, further in view of Park, and further in view of IBM TDB ("Method and System for Visually Constructing Document Type Definitions and Related Artifacts Using a Reusable Object Model," 2001, hereafter IBM TDB).

Claim 21 has been canceled, rendering moot the rejection of this claim.

Claims 11 and 32 depend from and include all of the features of Claims 1 or 22. Claims 11 and 32 have not been addressed separately but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Reconsideration thereof is respectfully requested.

#### **Claims 12 and 14-20**

Claims 12 and 14-20 have been canceled, rendering moot the rejection of these claims.

**IV. Additional Amendments**

Claims 45-52 have been newly added by the present Response. Applicant respectfully requests that new Claims 45-52 be included in the Application and considered therewith.

**V. Conclusion**

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: December 29, 2008

By: /Nathan L. Feld/  
Nathan L. Feld  
Reg. No. 59,725

Customer No. 80548  
FLIESLER MEYER LLP  
650 California Street, 14<sup>th</sup> Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800